中国政府机构上网指南

中國檢察出版社

中国政府机构上网指南

《中国政府机构上网指南》编委会

下 卷

中国检察出版社

第一章　网络安全基本概念

除非你试图保护的计算机是在一间上了锁的房间里，那里的访问受到控制，并且不存在来自室外与该计算机之间的连接，否则你的计算机就处于危险之中。全世界几乎每天都有闯入（break－in）和安全侵犯（security violation）行为发生。这些侵犯者并不仅仅是 Internet 的破坏者，他们包括为了个人目的或恶意地偷窃计算机时间或服务的雇员。

本章详细分析了计算机安全。你将学到什么是计算机安全，如何才能保护自己，以及使用哪些工具来帮助你执行这些安全任务。因为 Unix 是 Internet 上主要的操作系统，所以本章主要讲述 Unix 安全问题。但是，这并不意味着其它操作系统没有安全问题。不考虑硬件和操作系统制造商的话，你一定要全面理解你所处的危险境地。

第一节　美国国防部的安全级别标准

根据美国国防部开发的计算机安全标准，可信任计算机标准评估准则（Trusted Computer Standards Evaluation Criteria）——桔黄皮书（Orange Book），一些级别被用于保护硬件、软件和存储的信息免受攻击。这些级别均描述了不同类型的物理安全、用户身份验证（authentication）、操作系统软件的可信任性和用户应用程序。这些标准也限制了什么类型的系统可以连接到你的系统。

注意：桔黄皮书自从 1985 年成为美国国防部的标准以来，一直没有改变过。它多年来一直是评估多用户主机和小型操作系统的主要方法。其它子系统，比如数据库和网络，也一直是通过桔黄皮书的解释来评估的，例如，可信任数据库解释（Trusted Database Interpretation）和可信任网络解释（Trusted Network Interpretation）。

一、D1 级

D1 级是可用的最低的安全形式。该标准说明整个系统都是不可信任的。对于硬件来说，没有任何保护可用；操作系统容易受到损害；对于用户和他们对存储在计算机上信息的访问权限没有身份验证。该安全级别典型地指像 MS－DOS、MS－Windows 和 Apple 的 Macintosh System 7．x 等操作系统。

这些操作系统不区分用户，并且没有定义方法来决定谁在敲击键盘。这些操作系统对于计算机硬盘上的什么信息是可以访问的也没有任何控制。

二、C1 级

C 级有两个安全子级别：C1 和 C2。C1 级，或称自选安全保护（Discretionary Security Protection）系统，描述了一个典型的 Unix 系统上可用的安全级。对硬件来说存在某种程度的保护，因为它不再那么容易受到损害，尽管这种可能性仍然存在。用户必须通过用户注册名和口令让系统识别他们自己。这种组合用来确定每个用户对程序和信息拥有什么样的访问权限。

这些访问权限是文件和目录许可权限（permission）。这些自选访问控制（Discretionary Access Controls）使文件或目录的拥有者或者系统管理员，能够阻止某个人或几组人访问那些程序或信息。但是，这并没有阻止系统管理帐户执行活动。结果，不审慎的系统管理员可能容易损害系统安全。

另外，许多日常系统管理任务只能由以 root 注册的用户来执行。随着现在计算机系统的分散化，随便走进一个组织，你都会发现两三个以上的人知道根口令，这已经是司空见惯的事。由于过去无法区分是 Doug 还是 Mary 对系统所做的改变，所以这本身就是一个问题。

三、C2 级

第二个子级别 C2 可用来帮助解决这些问题。除 C1 包含的特征外，C2 级还包含其它的创建受控访问环境（controlled－access environment）的安全特征。该环境具有进一步限制用户执行某些命令或访问某些文件的能力，这不仅基于许可权限，而且基于身份验证级别。另外，这种安全级别要求对系统加以审核（audit），这包括为系统中发生的每个事件编写一个审核记录。

审核用来跟踪记录所有与安全有关的事件，比如那些由系统管理员执行的活动。审核还要求身份验证，因为没有它，如何能够确定实际执行命令的人就是某人呢？审核的缺点在于它需要额外的处理器和磁盘子系统资源。

使用附加身份验证，对于一个 C2 系统的用户来说，没有根口令而有权执行系统管理任务是可能的。这使得追踪与系统管理有关的任务有了改观，因为是单独的用户执行了工作而不是系统管理员。

这些附加身份验证不能与可应用于程序的 SGID 和 SUID 许可权限相混淆，而且它们是允许用户执行特定命令或访问某些核心表的特定身份验证，例如，那些无权浏览进程表的用户，当执行 ps 命令时，只能看到它们自己的进程。

四、B1 级

B 级安全包含三个级别。B1 级或标志安全保护（Labeled Security Protection），是支持多级安全（比如秘密和绝密）的第一个级别，这一级说明一个处于强制性访问控制之下的对象，不允许文件的拥有者改变其许可权限。

五、B2 级

B2 级，叫做结构保护（Structured Protection），要求计算机系统中所有对象都加标签，而且给设备（如磁盘、磁带或终端）分配单个或多个安全级别。这是提出较高安全级别的对象与另一个较低安全级别的对象相通讯的第一个级别。

六、B3 级

B3 级或安全域级别（Security Domain），使用安装硬件的办法来加强域，例如，内

存管理硬件用于保护安全域免遭无授权访问或其它安全域对象的修改。该级别也要求用户的终端通过一条可信任途径连接到系统上。

七、A 级

A 级或验证设计（Verify Design）是当前桔黄皮书中的最高安全级别，它包含了一个严格的设计、控制和验证过程。与前面提到的各级别一样，这一级包含了较低级别的所有特性。设计必须是从数学上经过验证的，面且必须进行对秘密通道和可信任分布的分析。可信任分布（Trusted Distribution）的含义是，硬件和软件在传输过程中已经受到保护，以防止破坏安全系统。

第二节　加拿大安全级别标准

加拿大政府已经着手设计它自己的可信任计算标准（trusted computing standard），这些标准包括两个部分：加拿大可信任计算机产品评估标准（Canadian Trusted Computer Product Evaluation Criteria，CTCPEC）和普通标准（Common Criteria）。

CTCPEC 提出了在开发或评估过程中产品的功能（functionality）和保证（assurance）。功能包括机密（confidentiality）、完整性（integrity）、可用性（availability）和可说明性（accountability）。保证集中于产品用以实理组织的安全策略的可信程度。

普通标准是美国、加拿大、法国、德国和英国联合调查与研究的结果，该文档包含了选择适当的 IT 安全措施的要求。

普通级别有 7 种保证级：EAL－1 至 EAL－7，以下几节分别讲述这些标准。

一、EAL－1

EAL－1 是最低的保证级别，它对开发人员和消费者来说是有意义的。它定义了最小程度的保证，并且是以对产品安全性能分析为基础的，它使用功能和接口设计来理解安全行为。

二、EAL－2

EAL－2 是在不需要强加给产品开发人员除 EAL－1 要求的任务之外的附加任务的情况下，可被授予的最高的保证级别。它执行对功能和接口规范的分析，以及对产品子系统的高级设计检查。

三、EAL－3

EAL－3 信述了一种中间的独立确定的安全级别，意味着安全由外部源来证实。该级别允许设计阶段给予最大的保证，而在测试过程中几乎不加修改。最大保证指的是在设计时已经考虑到了安全问题，面不是设计完之后再实现安全性。开发人员必须提供测试证据，包括易受攻击的分析，它们可以有选择地加以验证。这也是包括配置管理评价的第一级别。

四、EAL－4

EAL－4 是改进已有生产线的可行的最高保证级别。一个保证级别为 EAL－4 的产品是一个在设计、测试和检查方面都井井有条的产品，它向消费者提供了最高的安全级

别，这是以良好的商业软件开发经验为基础的。使用这些经验可以帮助排除困扰项目的一般软件设计问题。

除了 EAL－3 级的内容之外，EAL－4 也包括对产品的易受攻击性进行独立的搜索。

五、EAL－5

EAL－5 级对现有的产品来说是不容易达到的，因为这必须有意为了完成该标准来设计和创建产品。这里开发人员必须应用商业软件开发经验及特殊的安全工程技术。该级别试用于那些在严格的开发方法中要求较高保证级别的开发人员和用户。在这一级别上开发人员也必须提出设计规范和如何在产品中从功能上实现这些规范。

六、EAL－6

EAL－6 级包含一个半正式的验证设计和测试主件，该级别包括 EAL－5 级的所有内容，另外还要求提出实现的结构化表示。另外，产品要经受高低设计检查，而且必须保证具有高度的抗攻击性能。EAL－6 级也保证在设计循环中使用结构化的开发过程、开发控制和配置管现控制。

七、EAL－7

EAL－7 级只用于最高级别的安全应用程序，那里违反安全的高度危险性证明了开发代价的合理性，当然这些代价是由消费者来承担的。该级别包含完整的独立和正式的设计检查，有一个验证、设计和测试阶段。开发人员必须测试产品的每一个方面，寻找明显的或隐藏的易受攻击的地方，这都可以由一个独立的源来全部加以验证。这是一个耗费精力和时间的过程，从思想的形成到产品的最终完成，评估代现商必须全身心地投人进去。

第三节　局部安全问题

除了其它组织开发的安全产品或规则之外，你必须能够解决那些局部的，成仅限于你的组织或组织内部某个部门的安全问题，这些局部安全问题包括安全策略和口令控制。

一、安全策略

开发反映你的站点安全性的策略时，可以采用两个主要观点。这两个主要的立场形成了所有其它与安全有关的策略的基础，并且控制着实现它们的过程。

“没有明确允许的就是禁止的”，这是研究安全问题的第一个态度。这意味着你的组织提供了一个明确的和记录在案的服务集合，所有其它的都在禁止之列。例如，如果你决定允许匿名 FTP 传输到一台特殊的机器或从其传输出来，但是拒绝 telnet 服务，就明确地用文档说明支持 FIP，拒绝 telnet。

另外一种思想是“没有明确禁止的就是允许的”，这意味着，除非你明确指出一种服务不可用，则所有服务都是可用的。例如，如果你没有明确说明禁止对一台给定主机的 telnet 会话，那么就必须允许之（但是，通过不允许对该 TCP/IP 端口的连接，你仍

然可以禁止该服务)。

不管你遵循哪种思想，定义一种安全策略背后的原因是，在一个组织的安全受到损害的情况下，必须决定应当采取什么行动。这种策略也损述了哪些行动是可以容忍的，哪些不行。

二、口令文件

防卫非授权访问系统的第一道防线是/etc/password 文件。不幸的是，它也经常是最薄弱的环节！口令文件包括一些行或记录，每行被“:”分成 7 个域，如下例所述：

chare：u7mHuh5R4UmVo：105：100：Chris Hare：/u/chare：/bin/ksh

username：encrypted password：UID：GID：comment：home directory：login shell

表 9.1－1　　解释了每个文件的内容和典型值/etc/passw 文件

域名	描述	典型值
用户名	该域识别系统用户，它主要是为了方便人们。在一台给定的机器上用户名必须是唯一的，而且最理想的情况是，在一个组织内部用户名是唯一的	chare rech brb markd
加密口令	该域包含或能够包含加密口令。如何对口令进行加密，将在本章稍后介绍	u7mHuh5R4UmVo x * NOLOGIN
UID	这是系统上用户的数字表示	0－60．000
GID	这是用户所属注册组的数字表示	0－60．000
注释	它包含了有关用户的信息，通常列出用户的命名，电话号码或其它信息	Chris Hare Ops Manager x273
主目录	这是用户放置注册信息的目录	/c/chare /usr/lib/ppp /ppp－users
注册外壳	这是用户启动用来与系统进行交互的注册外壳。记住，并非所有系统外壳部是交互式的	/bin/sh /bin/esh /biu/ksh /bin/tcsh /usr/lib uucp/uucico /usr/lib /ppp/ppd/

表 9.1－1 提供了一些值得一提的附加信息。实际的加密口令并非一定要放在加密口令域中，该域可以包含其它值。例如，要防止某人使用一个特定帐户注册，口令可改变为一个无法匹配的值，如 NOLOGIN。但是，该域典型地包含一个 x 或“*”，指明该口令要么存储在可信任计算基础（Trusted Computing Base）中，要么在影像口令文

件中。

口令文件的许可权限是只读的，这意味着无人能够编辑该文件。类似地，影像口令文件和 TCB 中的文件一般也是只读的。

这些文件中的口令信息可通过 password 命令来修改。password 命令的许可权限包括 SUID（Set User ID）位，使用户像程序的拥有者一样执行该程序。由于 SUID 位的应用，用户能够改变口令，即使他无权编辑该文件。

三、影像口令文件

不包含来自 SecureWare 的高级安全选择的 Unix 版本可支持影像口令文件。当你在口令域中看到字符 x 时，那么用户实际的口令就存储在影像口令文件/etc/shadow 中。与/etc/passwd 文件不同，影像口令文件必须只有通过根才是可读的，如下例所述：

```
# ls - l /etc/shadow
- r - - - - 1 root auth 861 Oct 24 11: 46/etc/shadow
#
```

注意：SecureWare 是一家专门从事网络安全、安全 Web 服务器和增强保密邮件（privacyenhanced mail）的软件公司。SecureWare 为 SCO 编写代码，使 SCO Unix 操作系统遵从 C2 级安全。有关 SecureWare 更多的信息可见 http: //www. secueware. com。

文件/etc/shadow 的格式与/etc/passwd 是一致的，也有 7 个由“:”分开的域，但是/etc/shadow 只包含用户名、加密口令和口令生命期信息，如下例所述：

```
# cat/etc/shadow
root: DYQC9rXCioAuo: 8887: 0: 0::
daemon: * :: 0: 0::
mmdf: MZ74AeYMs4sn6: 8842: 0: 0::
ftp: b80lug/921MeY: 8363::::
anyone:: 8418::::
chare: 7xqmj9fj3bVw2: 9009::::
pppdemo: 4QYrWsJEHc81A: 9062::::
#
```

文件/etc/shadow 是由命令 pwconv 在 SCO 和 SVR4 系统上创建的。只有超级用户能够创建影像口令文件。在 SCO 系统上/etc/passwd 中的信息和被保护的口令数据库用于创建影像口令文件。在 SVR4 系统上，使用/etc/passwd 中的信息。

使用影像口令文件的主要优点是，它将加密口令放在一个普通用户无法访问的文件中，因而减少了破坏者窃取加密口令信息的机会。

四、拨号口令文件

直接在 Unix 系统上支持拨号访问的问题是有争论的。许多人仍然这么做，但一般来说它会引起问题。允许破坏者“侵入”系统可能会导致系统的损害。许多 Unix 系统提供了匿名 UUCP 访问，这会导致口令文件的丢失，而对组织不利。

在 Unix 系统上提供拨号访问的另一种方法是，通过一台支持身份验证的终端服务器来提供访问。在这种方式下，用户在允许网络访问之前，必须由终端服务器证实为合法。因为没有口令文件可从终端服务器上偷取，所以攻击终端服务器的任务变得更加困难了。

在那种你必须提供拨号访问 Unix 服务器的情况下，可以采取一些步骤来更好地保护你自己免受非授权网络访问之害。

遗憾的是，在串行端口线路上安装附加口令的能力不是所有 Unix 版本都具有的，它们最突出地可在基于 SCO 的系统中发现。对这些串行线路的拨号口令保护是通过/etc/dialups 和/etc/d－passwd 两个文件来控制的。文件/etc/dialups 包含一个由拨号口令保护的串行端口列表，该文件如下所述：

```
# cat dialups
/dev/tty2A
#
```

文件/etc/d-passwd 用于识别注册外壳。与使用用户的注册名的常规口令不同，拨号口令是与注册外壳相联系的。这意味着，使用 Bourne 外壳的每一个用户有相同的拨号口令。下述例子描述了典型的拨号口令：

```
# cat/etc/d－passwd
/bin/sh::
/usr/lib/uucp/uucico::
#
```

该文件中的每一行或记录包含两个由“:”分开的域。第一个域识别给定口令支持的外壳，第二个域列出了口令。在上面的例子中，两个外壳都没有口令，这意味着，用户注册时，不会被提示输入口令。

拨号口令是通过在 passwd 命令中使用－m 选项来增加的，该选项通知 passwd，搜集的口令支持拨号口令文件中的某个特定外壳。该命令的语法格式如下：

```
passwd －m shell-name
```

该命令的执行如下所述：

```
# passwd －m/bin/s:
Setting modem password for login shell: /bin/sh
Please enter new password:
Modem password: testing
Re－enter password: testing
#
```

在上例中，系统管理员为/bin/sh 外壳增加拨号口令，这意味着，注册到该系统并将 Bourne 外壳作为其注册外壳的任何用户，都会被提示输入拨号口令。它们必须输入的口令在上例中显示出来。与通常的 passwd 命令一样，实际的口令在其键入时并不显示。这里显示不出来，只是为了说明方便。注意，只有系统管理员能够改变一个外壳的拨号口令。

passwd 命令改变 d-passwd 文件的内容，将新的口令包括进去，如下所述：

```
# cat d - passwd
/bin/sh: ORa691. Na1jjQ:
/usr/lib/uucp/uucico::
#
```

如上例所述，现在 Bourne 外壳用户在注册到文件/etc/dialups 中指定的终端端口时，必须提供附加口令。

下面的例子描述了现在当用户使用拨号口令注册时经历的过程：

```
gateway
gateway! login: chare
Password
Dialup Password:
Last successful login for chare: Fri Oct 28 22: 52: 02 EDT 1994 on tty2a
Last unsuccessful login for chare: Tue Oct 18 16: 27: 56 EDT 1994 on ttypl
SCO Unix System V/386 Release 3. 2
Copyright (C) 1976 - 1989 UNIX System Laboratories, inc.
Copyright (C) 1980 - 1989 Microsoft Corporation
Copyright (C) 1983 - 1992 The Santa Cruz Operation, Inc.
All Rights Reserved
gateway
TERM = (ansi)
#
```

如上所见，直到用户输入用户名和口令，才通过通常的注册过程。这些被验证有效之后，检查拨号口令控制文件/etc/dialups。当用户连接的终端放置在该文件中，且注册外壳是 Bourne 时，提示用户输入拨号口令。如果拨号口令输入正确，就授权用户访问该系统，如上例所述。但是如果拨号口令错误，就放弃注册，用户被迫重新启动，如下所述：

```
gateway
gatewayllogin: chare
Password:
Dialup Password:
Login incorrect
Wait for login retry:
login: chare
Password:
Dialup Password:
Last Successful login for chare: Fri Oct 28 23: 28: 28 EDT 1994 on tty2a
I unsuccessful login for chare: Fri Oct 28 23: 30: 55 EDT 1994 on tty2a
SCO Unix System V/386 Release 3. 2
Copyright (C) 1976 - 1989 UNIX System Laboratories, Inc.
```

Copyright (C) 1980 - 1989 Microsoft Corporation
Copyright (C) 1983 - 1992 The Santa Cruz Operation, Inc.
All Rights Reserved
gateway
TERM = (ansi)
$

五、组文件

文件/etc/group 用来控制访问那些不是用户所拥有的文件。让我们回忆一下许可权限是如何工作的：如果用户不是文件的拥有者，那么就检查用户所属的组，查看用户是否是拥有该文件的组的成员。组员列表存储在/etc/group 中。该文件的格式如下所述：

tech: *: 103: andrewg, patc, chare, erict, lorrainel, lens
group name: password: GID: userlist

表 9.1 - 2 解释了文件/etc/group 中包含的第一个域。

表 9.1 - 2　　文件/ect/group

域名	描述	例子
组名	使用组名主要为了人们的方便	tech sales group
口令	切换到该组时要使用的口令	*
GID	标记下所有文件上的数字组的 ID 号码	0 - 30、000
用户列表	由“,”分开的组员列表	chare andrewg

组文件的口令是不使用的，也没有存易的机制用于在文件中安装口令。如果用户试图切换到它们不是其成员的组，就会被提示储人口令，如下所述：

$ newgrp tech
newgrp: Password
newgrp: Sorry
$ grep tech /etc/group
tech:*: 103: andrewg, patc
$

如果执行该命令的用户已经成为组 tech 的成员，那么 newgrp 命令就会成功。但是许多 Unix 的当前版本使用户能够同时成为多个组的成员，因此就会减少或取消使用 newgrp 命令需要。

考虑到下面一点是很重要的：Berkeley Unix 使用 Wheel 组深人保护了根帐户。只有那些也是 wheel 组成员的用户可以使用 su 命令来改变根。

提示：要不提供实际的口令而访问根注册，系统管理员应该考虑使用 sudo 命令，

有关 sudo 的更多的信息可从 http//www. cs. colorado. adu/ – millert/sudo/找到。

第四节　口令生命期和控制

许多 Unix 版本都提供了口令生命期（password aging），该机制控制用户何时可以加密口令后通过在口令文件中插入一个值来修改他们的口令。该值定义了用户修改口令之前必须经过过的最小时间间隔和口令有效期满之前可以经历的最大时间间隔。

上述解释用数轴来表示将会更加清楚，如图 9.1 – 1 所示。

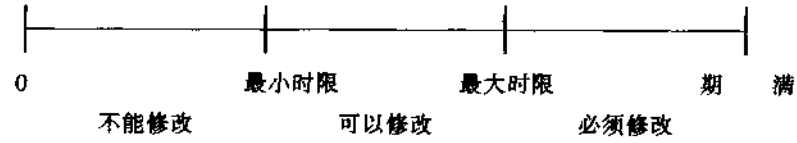

图 9.1 – 1　口令生命周期

口令生命期控制信息与加密口令存储在一起，以一系列可打印字符的形式出现。控制包含在口令之后，用逗号分开。通常，逗号后面的字符表示下述信息：

1. 口令有效的最大周数。

2. 用户可以再次改变其口令之前必须经的最小周数。

3. 口令最近改变的时间。

表 9.1 – 3 定义了口令生命期的值。

表 9.1 – 3　　口令生命期值

字符	值（单位：周）	字符	值（单位：周）
	0		1
0	2	1	3
2	4	3	5
4	6	5	7
6	8	7	9
8	10	9	11
A	12	B	13
C	14	D	15
E	16	F	17
G	18	H	19
I	20	J	21
K	22	L	23
M	24	N	25

字符	值（单位：周）	字符	值（单位：周）.
O	26	P	27
Q	28	R	29
S	30	T	31
U	32	V	33
W	34	X	35
Y	36	Z	37
a	38	b	39
c	40	d	41
e	42	f	43
g	44	h	45
i	46	j	47
k	48	l	49
m	50	n	51
o	52	p	53
q	54	r	55
s	56	t	57
u	58	v	59
w	60	x	61
y	62	z	63

使用上述信息查看口令的生命期，你可以破译下述例子的含义：

chare：2eLNss48eJ/GY，C2：215：100：Chris Hare/usr1/chare：/bin/sh

在上例中，口令已经被设置了生命期，使用“C”值定义了口令到期前的最大周数，“2”定义了用户能够再次更改口令前必须经过的最小周数。每次用户注册时，就将上次改变的值与最大值作比较，来检查该口令的生命到期了没有。例如，如果用户必须每三个星期更改一次口令，而三周以前口令已经被更改过，那么用户就必须修改其口令。当用户改变其口令时，上次修改的值就被更换，以反映它是何时被修改的。

生命期控制机制识别两种特殊的情况：一种迫使用户在下次注册时改变其口令；一种禁止用户修改口令。

要强迫用户修改其口令，比如一个新用户，该用户的口令域要修改为包含一个“，”后跟两个时间，表示最大和最小时间间隔，在这种情况下，用户在下次注册时被迫修改其口令。一旦修改之后“强迫”控制信息就从口令项中删除了，下面显示了一个口令中强迫项的例子：

chare：2eLNss48eJ/GY，/：215：100：Chris Hare：/usr1/chare：/bin/sh

第二种特殊的情况禁止用户修改其口令。这种情况是通过设置最大值小于最小值来建立的（也就是说，第一个值小于第二个值）。在这种情况下，用户被告知其口令不能改变，如下所述：

chare：，..：215：100：Chris Hare：/usr1/chare：/bin/sh

如果使用当前市面上的更新、更安全的 Unix 版本，你可能会听到术语口令生存期（password lifetime）。它是最大时限之后用户仍然能够使用到期的口令注册其帐户的宽

限期。当生存期到期时，该帐户就被取消了。当用户试图使用一个取消的帐户注册到系统时，它就会被告知，该帐户已经取消，请与系统管理员联系。

口令生命期机制并没有禁止用户改变其口令然后再将其改变回来。只有一些 Unix 系统版本追踪用户使用了什么口令。实现口令生命期的实际过程是依赖于版本的。要在你的系统上实现它，请查看系统文档。

列表 9.2－1 中的程序 Pwexp. pl 是一个 PERL 语言程序，它在用户的口令即将到期时指导用户，以便他们在系统通知他们其口令已经到期的时候能够做好准备。但是，注意，该程序的版本是面向标准 System V Unix 的，没有使用影像口令文件。

注意： PERL（Practical extraction and Report Language），即实用析取和报告语言，是一种免费的广泛使用的编程语言，它将多个 Unix 命令元素结合成一个整齐的包，一些特征包括模式匹配、数组、条件编程和子例程。PERL 可从许多匿名 FTP 站点获得，包括 uunet. uu. net 和 netlabs. com。

该程序提出了在 System V 版本 3. 2 的以下的 Unix 上实现的生命期机制。在此水平上，为了增加系统安全性，做了一些改动。AT&T/USL 版本转去使用/etc/shadow 文件存储口令信息，而 SCO 的实现则使用 Secure Ware 公司的可信任计算基础（Trusted Computing Base）实用程序。在 SCO 的 Unix 3. 2v4 产品中，生命期控制信息可能在 /etc/passwd、/etc/shadow 中，也可能在可信任计算基础文件中，这依赖于你配置何种级别的安全。记住下面一点也是很重要的，在 SCO 中，保护口令数据库（Protected Password Database）和/etc/shadow 数据库可以同时使用。

第五节　破坏者和口令

术语“黑客”（hacker，也叫计算机迷），并不总是具有这样一种反面的能义。确切地说，它是这样一个术语，是指那些固执的、试图破坏事物、弄清它们工作原现的人。正是由于这种声誉，而且大多数从事这项工作的人都是计算机科学的奇人，所以术语才带有否定的含义。但是，由于我了解一些“较好的”黑客，在本章的讨论中，那些坏孩子被称作破坏者（vandal）。

破坏者出于种种原因希望访问你的系统，这些原因可能是：

1. 仅仅为了好玩
2. 浏览观光
3. 偷窃计算机资源，如 CPU 时间
4. 窃取商业秘密或其它有价值的信息

注意： 并非每一次试图对你的系统的访问都是意欲伤害你的系统。但是，在大多数情况下，应该这样看待这些访问。不考虑攻击背后的原因，破坏者最需要的信息是文件

/etc/passwd。当破坏者拥有有效用户的帐户名列表时，创建猜测口令的程序就是小菜一碟了。但是，许多现代注册程序在注册提示之间包含一个时间延迟，随着每一次不成功的注册尝试，该延迟变得越来越长。它们也会包含程序代码在记录了太多的不成功的尝试情况下取消访问端口。

主要的保护员是使用/etc/shadow 或保护口令数据库，因为这些文件和目录要求根访问来浏览它们，这使得破坏者要获取加密口令信息更加困难。然而回忆一下，/etc/passwd 中的口令信息是加密的。当破坏者无法解密存储的口令时，他怎储希望获取访问权呢?

理解破坏者加何破坏口令

那种口令被加密后就不储被解密的说法是正确的，但是这并不意味者口令仍就是安全的。口令网入者（cracker）是破坏者使用的一个程序，它试图通过将它们与字典中的单词相比较来“储储”文件/etc/passwd 中的口令。闯入者程序的成功依赖于 CPU 的资源、字典的质量和用户拥有/etc/passwd 的备份的事实。

口令闯入者编写起来相当容易。一个简单的尽管是效率不高的闯入者，可以用大约 60 行的 C 代码或 40 行的 PERL 代码储写出来。就这么简单。为了对你的系统提供一个更好的口令，系统管理员完全可以考虑编写一个。但是，应该提出警告的是，你也许正在邀请灾难到来。如果程序是足够有效的，或者检测到了你的系统口令中的一些缺陷，你也许能够进一步击败你的机器的安全性。一个高效的闯入者程序也许被别人信去，然后用于访问其它的机器。其它更好的保护系统的方式是使用逻辑安全。

而且，由于你的闯入程序存在被信取的可能，如果由于使用该程序而产生了损害，就会引起一些法律问题，这种问题通常不会是很轻微的。这是这种问题能产生的一个很严重的例子。

例如，系统管理员通常都遇到这样一个很不幸的情况：忘记了一台机器的根口令且没有重新安装的分布媒介。在这种储况下，他就是一个友好的破坏者，因为他拥有这台机储。他通过匿名 UUCP（Unix to Unix Copy）检索/etc/passwd 文件（这是为了安全的需要）。当拥有该文件之后，他将它发送给试图使用口令闯入程序的熟人。在不成功的信况下，他将它发送至另外一台机器，在那里超级计算机努力奋战了 18 个小时而终于获得了根口令。最滑稽的是该口令竟然是它工作的公司的名字。

有时这种简单的方法竟然如此的有效。在过去几年中，一些口令闯入或猜测程序已经被邮寄到 Internet，这并非意味着你应该拥有一个。事实上，你可能并不想在你的系统上运行这种程序。这种程序的作者在它们的文档中滑楚地说明，他们假定没有责任信用这些程序，尽管他们声明这些程序是非常高效的。

我们郑重建议你拥有一个任何口令闯入程序的复本，并经常使用它们。但是，要保证公司的安全策略，允许你（系统管理员）使用该文件，同时禁止其信任何人这么做。

你可以使用一些口令闯入工具。

这个故事的寓义在于把试图猜测口令的程序认为是危险的且不是过去常常能够被克服的潜在的问题。

警告： 如果你选择拥有审核和其它评估可用的安全程序，那么它们不包含在一个可访问的网络系统中是必要的。使它们在网络中可用可能比你自己小心地拥有它将引起更大的悲伤。

第六节 C2 安全性和可信任计算基础

可信任计算基础（TCB）是 C2 级 Unix 系统的安全系统的一部分，它为系统的操作和管理增添了明显的复杂性。该系统将/etc/passwd 文件的位移到其它地方，并为原始信息增加附加信息，组成可信任计算基础数据库的文件分散在几个不同的目录等级结构中。但是编辑这些文件并不是一个好主意，因为这会导致对你的系统的严重损害。

在一个使用 TCB 的系统上，“*”被放置于/etc/passwd 的口令域，这是因为实际的用户口令是和可信任计算基础中的其它用户信息一起存储的。使用 TCB 并没有改变系统的操作，在一些 Unix 版本中，比如在 SCO Unix 中，即使你没有使用 C2 安全性，可信任计算基础仍然用于提供安全服务。

表 9.1－4 显示了可信任计算基础的六个组件

表 9.1－4 可信任计算基础

组件	描述
/etc/passwd	系统口令文件
/tcb/autb/files/	保护口令数据库
/etc/auth/systems/ttys	终端控制数据库
/etc/autb/systems/files	文件控制数据库
/etc/autb/subsystems/	保护子系统数据库
/etc/autb/system/default	系统缺省数据库

当引用可信任计算基础时，它指的是表 9.1－4 中的所有组件而不是任何一个单个组件。

一个可信任系统的操作引出了一些概念，这是为了防止系统处于危险境地而必须理解的。

可信任计算基础是由包含 Unix 核心和维护可信任计算基础的实用程序的软件的集合组成。这些实用程序包括用于验证和改正口令数据库中问题的 authck，和验证系统文件的准确性的 integrity。可信任计算基础实现了系统的安全策略，该策略是一个控制主题（subject，如进程）和对象（Object，如文件、设备和过程中通讯对象）之间的交互的操作规则的集合。

只有当一个动作可被追溯至个人时，才可定义该动作的可说明性（accountabibty）。在传统的 Unix 系统上，不止一个人知道根口令，对于该动作来说要追溯至任何个人，

如果不是不可能的话，至少也是困难的。像 cron 和 lp 这样的伪帐户是匿名运行，它们的行为只有当系统信息被改变之后才可被追踪。这在可信任 Unix 系统上得到了修正，因为每个帐户都与一个真实的用户联系在一起，每个动作都是经过审核的，并且每个动作都与某个特定的用户相联系。

传统的 Unix 系统几乎不进行识别和身份验证（Identification and Authentication，I&A）。在传统的 Unix 上，用户通过提供注册名和口令的组合来注册，它是通过搜索文件/etc/passwd 来确认有效的。如果注册名和口令是正确的，那么用户就被允许访问该系统。在一个可信任系统中，使用一些附加的规则来改进标准的 I&A 技术，例如，创建了一些新的修改和产生口令的过程，对口令数据库的各个部分提供了更好的保护，以防止别人偷窃访问。

下面的例子描述了一个 SCO 系统上的可信任计算基础中的典型用户项，它详细提供了一个特定用户的信息。该信息永远不应该被手工编辑，因为这样做可能使你的系统处于无用状态。例子如下所述：

```
chare:  u-name = chare:          # Actual user name
      : u-id # 1003:  \          # User ID
: u-pwd = MWUNe/91rPqck:  \ # Encrypted password
: u-type = general:  \           # User Type
: u-succhg # 746505937:  \       # Last Successful Password Change
: u-unsucchg # 746506114:  \     # Last Unsuccessful Password Change
: u-pswduser = chare:  \         #
: u-suclog # 747066756 \         # Last successful login
: u-suctty = tty02:  \           # Last Successful login on tty
: u-unsuclog # 747150039:  \     # Last unsuccessful login
: u-unsuctty = tty04:  \         # Last unsuccessful login on tty
: u-numunsuclog # 1:  \          # Number of unsuccessful logins
: u-locka:  \                    # Lock Status
: chkent:                        #
```

上述例子已经被修改了，在项目中插入了注释。“#”后面的文本不出现在文件中。这里包含此例是为了说明，在 TCB 中事实上也追踪了其它信息，这并不是所有的信息，而只是包含在一个文件中的内容。对每一个用户来说，以用户名命名的文件被存储起来，并且包含上例中所描述的信息。

在以上说明中，项 u-succhg 显示值为 746505937，这是 Unix 追踪的时间。该值是从 1970 年 1 月 1 日开始计算的秒数。该值可以提供给 Unix 核心中的一个将其转换为实际的日期和时间的函数。

传统的 Unix 系统保存了有关系统活动的有限的信息，在某些情况下，只有当它们被这样配置时它们才这么做。在可信任的 Unix 中，审核是保证动作与特定用户相联系的主要特征。审核系统为每个动作编写一系列的记录，以产生有关系统上发生的事件审核踪迹（trail）。该审核踪迹由每个主题和对象之间的动作组成，这些动作是关于对象访问、对主题和对象的修改，以及系统特征的，它们要么成功，要么不成功。这样，审